TEAPC: Adaptive Computing and Underclocking in a Real PC

Augustus K. Uht and Richard J. Vaccaro
Dept. of Electrical and Computer Engineering

Copyright © 2004, A. K. Uht, R. J. Vaccaro
Patent applied for.
Background

Prior work: **TEAtime (Timing Error Avoidance)**

- “Maximized” performance. All hardware. Method:
  - Use one-bit copy of worst-case delay path in system; add a small safety-margin delay.
  1. *Speed up clock ‘til just before real error would occur*
  2. *Slow down clock*
  3. *Repeat: GOTO 1.*

- TEAtime adapts to:
  - Current environmental conditions (e.g., temperature)
  - Current operating conditions (e.g., voltage)
  - Prior manufacturing conditions (quality of a prod. Run)

- Prototype almost doubled performance
Outline

1. Motivation and Goals
2. Related Work
3. TEAPC’s Features
4. Feedback-Control System
5. TEAPC Software and Hardware Details
6. Experiments
7. Summary
8. Demo
Motivation and Goals

- **Motivating Goals:**
  - Realize TEAtime characteristics in a real computer
    - Adaptive computing
    - Improved performance – “Better-than-Worst-Case”

- **Additional Goals:**
  1. **Workload adaptation**
  2. **Reduced power consumption**
  3. **Improved reliability**
  4. **Disaster tolerance (always enabled)**
  5. …and all in a real machine

- **BUT:** can’t redesign or build Pentium 4’s
- **SO:** use real IBM/Intel-standard PC
Related Work

- Rohou & Smith, 1999 – temperature adaptive system
  - Adjusted temperature with frequency changes
  - BUT: required modifying OS (Linux)
  - Performance not enhanced
- Skadron et al, 2002 – temperature adaptive system
  - Used classical feedback control theory
  - Modeled and controlled temperatures of parts of a chip
  - Instruction-fetch toggling controlled temperature
  - Performance reduced
- (Note: ~all adaptive methods useful for either or both:
  - Performance improvement
  - Power reduction)
TEAPC’s Key Features

• High thermal capacitances and delays, hence:
  Modern feedback-control theory and system used
  – Input: CPU’s internal temperature (from embedded thermal diode)
  – Outputs: CPU’s clock frequency (I/O clocks unmodified), Vcore

• User-control provided by control system’s Tset point

• Entire system realized in Windows application
  – No OS modifications – Windows 2000 used
  – No hardware modifications – all COTS parts

• Applicable to many kinds of PC’s
  – New designs of commercial PC’s
  – Possibly existing motherboards (MOBO)
Control System History

1. TEAtime approach
   - Direct feedback
   - In TEAPC: Oscillated

2. TEAPC-0
   - External sensors
   - Too slow
   - Uncontrollable

Note: all of the constants of the four components differ.
Final Control System

- Only input is CPU temperature (feedback line)
- Primary output is CPU frequency (N) \[ \text{sometimes: } V_{\text{core}} = f(N) \]
- State-space discrete control system design (modern)
- Quick response

Clock Synthesizer, CPU and CPU internal sensor

\[
\begin{align*}
T_{\text{set}} (^\circ C) + & \quad \Delta^\circ K \\
\rightarrow & \quad K_{NT} \quad N/^\circ K \\
& \quad 17.4 \\
\rightarrow & \quad \Delta N \\
\rightarrow & \quad K_G \\
& \quad 0.28 \\
\rightarrow & \quad \Delta N \\
\rightarrow & \quad 1/s \\
\rightarrow & \quad + \\
\rightarrow & \quad N \\
\rightarrow & \quad \Delta N \\
\rightarrow & \quad K_{fN} \quad \text{freq}/N \\
& \quad 8.26e6 \\
\rightarrow & \quad B \\
& \quad \frac{\text{freq}}{s + A} \\
& \quad B=0.0364 \\
& \quad A=0.03345 \\
\rightarrow & \quad K_{Tf} \quad ^\circ C/\text{freq} \\
& \quad 6.900e-9 \\
\end{align*}
\]
teapc Control Program

- Windows application – no changes to OS
- Uses x86 I/O address space to access hardware
- Small: 800 kilobytes
- Fast: < 5% CPU utilization
- Control loop updated every second
- Hard limits on max./min. frequency
# TEAPC Components

<table>
<thead>
<tr>
<th>PC Component</th>
<th>Manufacturer</th>
<th>Part Number/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motherboard</td>
<td>Gigabyte</td>
<td>GA-8KNXP (Rev. 2); w/DPS regulator</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel</td>
<td>P4 3.0 GHz, 800 MHz bus</td>
</tr>
<tr>
<td>Chipset</td>
<td>Intel</td>
<td>875P. ICH5R</td>
</tr>
<tr>
<td>Clock Synthesizer</td>
<td>ICS</td>
<td>ICS952635</td>
</tr>
<tr>
<td>Super I/O (Environment Mon.)</td>
<td>ITE</td>
<td>IT8712F V0.6</td>
</tr>
<tr>
<td>CPU Volt. Regulator Control</td>
<td>ITE</td>
<td>IT8206R V0.1</td>
</tr>
<tr>
<td>Main Memory</td>
<td>Ultra</td>
<td>U10-5903R; 2 x 512 MB; 400 MHz DDR, Dual Channel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Operated at 320 MHz.)</td>
</tr>
<tr>
<td>Operating System</td>
<td>Microsoft</td>
<td>Windows 2000 SP4, HT disabled</td>
</tr>
<tr>
<td>Disk System – RAID 0+1</td>
<td>ITE</td>
<td>GigaRAID IT8212F</td>
</tr>
<tr>
<td>Disks</td>
<td>Maxtor</td>
<td>4 x 6E040L0, 40 GB, 133MHz IDE</td>
</tr>
<tr>
<td>Equipment for experiments only</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fan Controller &amp; Temp. Mon.</td>
<td>Thermaltake</td>
<td>Hardcano 12; for 4 fans, 4 thermocouples</td>
</tr>
<tr>
<td>Power Meter</td>
<td>Electronic</td>
<td>watts up? PRO</td>
</tr>
<tr>
<td></td>
<td>Educational</td>
<td>(Note: this is the unit’s model name.)</td>
</tr>
<tr>
<td></td>
<td>Devices</td>
<td></td>
</tr>
<tr>
<td>CPU Fan Controller</td>
<td>custom</td>
<td>On/Off, control sel. (MOBO or Hardcano)</td>
</tr>
</tbody>
</table>

---

**Note:**
- **Motherboard:** Gigabyte GA-8KNXP (Rev. 2); w/DPS regulator
- **CPU:** Intel P4 3.0 GHz, 800 MHz bus
- **Chipset:** Intel 875P, ICH5R
- **Clock Synthesizer:** ICS ICS952635
- **Super I/O (Environment Mon.):** ITE IT8712F V0.6
- **CPU Volt. Regulator Control:** ITE IT8206R V0.1
- **Main Memory:** Ultra U10-5903R; 2 x 512 MB; 400 MHz DDR, Dual Channel (Operated at 320 MHz.)
- **Operating System:** Microsoft Windows 2000 SP4, HT disabled
- **Disk System – RAID 0+1:** ITE GigaRAID IT8212F
- **Disks:** Maxtor 4 x 6E040L0, 40 GB, 133MHz IDE
- **Equipment for experiments only:**
  - **Fan Controller & Temp. Mon.:** Thermaltake Hardcano 12; for 4 fans, 4 thermocouples
  - **Power Meter:** Electronic Educational Devices watts up? PRO (Note: this is the unit’s model name.)
  - **CPU Fan Controller:** custom On/Off, control sel. (MOBO or Hardcano)
Underclocking & Performance “Maximization”

CPU nominal freq. = 3.0 GHz →
freq., Vcore **unlinked** < Load Adaptation > freq., Vcore linked

![Graph 1: CPU Freq. & PC Total Power](image)

- **Freq. (GHz)**
- **Vcore (V)**
- **Power (100 W)**

![Graph 2: Temperature vs. Time](image)

- **Tset**
- **Traw**
- **Tavg**
- **(all deg. C)**
Disaster Tolerance

- Example: CPU Fan dies….
- Changes (automatic, via feedback system):
  - Freq: 3.5 GHz -> 1.1 GHz
  - Vcore: 1.5125 V. -> 1.0875 V.
  → Power: 222 W. -> 140 W. (37% savings)
- CPU temperature stabilizes at safe value (with this CPU)
- System still functional
Summary

- TEAPC realizes:
  1. Better-than-worst-case performance
  2. *Adaptive* operation to both environment and loading
  3. Low-power, high-reliability operation
  4. Disaster tolerance

- Feedback-control great for a system, too
- **Underclocking** is a great tool
- **It Works!**
…and now it’s time for the:

DEMO
TEAPC:
Adaptive Computing and Underclocking in a Real PC

Augustus K. Uht and Richard J. Vaccaro
Dept. of Electrical and Computer Engineering

Copyright © 2004, A. K. Uht, R. J. Vaccaro
Patent applied for.
TEAtime Block Diagram

- Timing Error Avoidance system
  - **Blue**: TEAtime hardware
  - **Green**: on FPGA

NOTES: - $w, x >> 1$; - DAC: Digital-to-Analog Convertor; - VCO: Voltage-Controlled Oscillator.
TEAPC Block Diagram

- **CPU** Intel P4
- **Northbridge** Intel 875P
- **Main Memory** 1 GB Dual Channel 400 MHz Ultra

- **Super I/O** ITE 8712F
- **Clock Synthesizer** ICS 952635
- **Clock** Synthesizer

Connections:
- FSB (Front Side Bus)
- LPC Bus (Low Pin Count)
- SMBUS - IIC Bus

Components:
- CPU Vcore Power Supply
- CPU Vcore Regulator Control

Miscellaneous:
- CPU Clock
- Memory Clock
- (Environment Monitor)

Note: Only directly relevant components and connections are shown.
Experiment Setup

[Image of computer setup with monitoring software and hardware components]
## Operating Scenarios

<table>
<thead>
<tr>
<th>Goal</th>
<th>Tset</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low power, high reliability</td>
<td>Low</td>
<td>Low CPU freq. &amp; Vcore. Un-intensive apps., e.g.: web-browsing. Still works.</td>
</tr>
<tr>
<td>Mid-power &amp; reliab.</td>
<td>Mid-range</td>
<td>Environment adaptivity</td>
</tr>
<tr>
<td>High performance</td>
<td>High: freq. pegged</td>
<td>High CPU freq. &amp; Vcore. Ex.: FPGA net routing; 3-D games.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Goal</th>
<th>Tset</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disaster tolerance</td>
<td>Any</td>
<td>TEAPC always enabled for disaster tolerance.</td>
</tr>
<tr>
<td>E.g.: High temps.: CPU temp. kept to safe level. Still works.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E.g.: Low temps.: High perf.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>